How to paint a tree

**Time Limit: 10000/5000 MS (Java/Others)    Memory Limit: 65536/32768 K (Java/Others)**

Description

There is a **binary** tree, whose vertexes are either black or white, and now YXH wants to paint the tree into only one color. So easy to paint every vertex one by one, so she is not willing to choose this way. She comes up with two ways to paint.

1: Choose two vertexes, then there will be only one shortest path between them, she reverses the color of every vertex on the path (black to white, white to black, including the two vertexes), **no two paths can overlap each other**;

2: Choose a subtree, reverse the color of all the vertexes on the subtree.

If it costs 1 second to finish any operate, she wants to know at least how long it will take to paint the tree into only one color. (It is obvious that in some situations she has to paint the vertex one by one to minimize the time)

Input

The input contains less than 200 cases, the first line of each case is N (N < 10000), the number of vertexes (1~N). Then n-1 lines, each line has two integers a, b, donating that there is an edge connecting a and b. Then one line has n numbers, 0 or 1, donating the color of the ith vertex;

The tree’s root is always vertex 1.

Output

For each case output one line, the minimum time to paint the tree into one color.

Sample input

6

1 2

1 3

3 4

3 5

5 6

0 0 1 1 1 1

4

1 2

2 3

2 4

1 0 0 0

Sample output

Case 1: 1

Case 2: 1

Hint

Sample1: you can choose the vertex 2 and 1 to reverse vertex on the path, you also can choose vertex 3 and reverse its sub tree, both of this operate can achieve the goal in 1 second.

Sample2: you can choose the vertex 4 and vertex 4 to reverse vertex on the path, you also can choose vertex 2 and reverse its sub tree, both of this operate can achieve the goal in 1 second.